

九州大学
情報基盤センター

広報

全国共同利用版
2003年 Vol. 3 No. 3

目次

研究開発

- 計算機クラスタを用いた並列メタ戦略アルゴリズムの
設計と実装 …………… 石橋正裕、小野廣隆、朝廣雄一、山下雅史 199
有限要素法による大規模音場解析
…………… 大鶴 徹、富来礼次、岡本則子、福田みどり 211

解説

- CVS でファイル管理 (2) …………… 池田大輔 216

報告

- 平成 15 年度講習会開催一覧…………… 228
お知らせ …………… 236
人事異動 …………… 241

- 編集後記 …………… 242

Computing and Communications Center
Kyushu University

センターアクセス一覧

インターネット

接続システム名	ドメイン名	IPアドレス
GP7000F/900	kyu-cc. cc. kyushu-u. ac. jp	133.5.9.1
VPP5000/64	kyu-vpp. cc. kyushu-u. ac. jp	133.5.8.160
端末サーバ	wisdom. cc. kyushu-u. ac. jp	133.5.8.1
GS320	kyu-ss. cc. kyushu-u. ac. jp	133.5.8.158

交換回線使用電話番号

接続システム	通信速度	制御手順	電話番号
端末サーバ	14400~64000 (ISDN回線, 一般電話, PIAFS)	PPP (同期) PPP (非同期)	092-642-7340

スーパーコンピュータ VPP5000/64 (kyu-vpp) の制限値

適用ジョブ	キュー	CPU時間	メモリサイズ		備考
			省略値	最大値	
バッチジョブ	s	1時間	2GB	7GB	非並列短時間
	s8		2GB/PE	7GB/PE	8並列短時間
	p1	20時間	2GB	14.5GB	非並列
	p8		2GB/PE	7GB/PE	8並列
	p16				16並列
	p32				32並列
	x8		10GB/PE	15GB/PE	8並列大容量メモリ
x16	16並列大容量メモリ				
TSSジョブ	—	1時間	1GB		ベクトル演算可

汎用UNIXサーバ GP7000F/900 (kyu-cc) の制限値

適用ジョブ	キュー	CPU時間	メモリサイズ	備考
バッチジョブ	sc	120時間	4GB	非並列向け
	sc8		8GB	8CPUまで使用可
	sc32		32GB	32CPUまで使用可
TSSジョブ	—	12時間	1GB	12CPUまで使用可

スカラー並列サーバ GS320 (kyu-ss) の制限値

適用ジョブ	キュー	CPU時間	メモリサイズ	備考
TSSジョブ	—	40時間	16GB	32CPUまで使用可

計算機クラスタを用いた 並列メタ戦略アルゴリズムの設計と実装

石橋 正裕 (Masahiro Ishibashi) *

小野 廣隆 (Hirotaka Ono) †

朝廣雄一 (Yuichi Asahiro) ‡

山下 雅史 (Masafumi Yamashita) †

* 九州大学大学院システム情報科学府 † 九州大学大学院システム情報科学研究院

*† Department of Computer Science and Communication Engineering,
Graduate School of Information Science and Electrical Engineering,
Kyushu University

‡ 九州産業大学情報科学部社会情報システム学科

‡ Department of Social Information Systems, Faculty of Information Science,
Kyushu Sangyo University

概要

工学分野を始めとする様々な分野において現れる問題の多くは現実的な時間で最適解を求めることが難しいとされる NP 困難のクラスに属しており、このため最適解ではないまでも比較的良い解を高速に求めることができるメタ戦略の研究が近年盛んに行なわれている。逐次アルゴリズムを計算機クラスタ上で並列化する場合、十分な並列度を保つような実装が求められるが、メタ戦略はその動作が複雑であるため、一般に逐次メタ戦略の並列化には対象となるメタ戦略の動作の詳細までの把握が必要とされる。本稿では対象とするメタ戦略の動作の詳細を知ることなく高性能な並列メタ戦略を設計することのできる、メタ戦略並列化手法を提案する。本手法はメタ戦略に対する特徴付といくつかの簡単な予備実験の結果解析に基づいており、有名な NP 困難問題の一つである一般化割当問題に対する実装例では、公開されているベンチマーク問題への最良解を更新するなど、十分な性能を持つことが確認された。

1 はじめに

近年、NP 困難な組合せ最適化問題に対する高精度近似解法であるメタ戦略が、その有用性から盛んに研究されている [3]。多くのメタ戦略は局所探索法に基づいており、現実的な時間で精度の高い近似解を求められることが知られている。一方、メタ戦略の発展に伴い、メタ戦略を複数の計算機で並列化することで、より高精度で高速な解法を開発しようとする研究も盛んに行われている [5]。メタ戦略を並列化する場合、対象とするメタ戦略を熟知した上でパラメータチューニングや実験的な作り込みが必要とされる場合が多い。しかし、このような並列化手法では、一般に個々のメタ戦略のデータ構造、動作の理解に多くの労力が必要となる。また、実際に実装して動かしてみるまでその性能が分からないため、使用計算機数を増やしても期待通りの性能が得られないなどの問題点がある。そこで本研究ではメタ戦略の並列化プロセス自体に着目し、与えられたメタ戦略アルゴリズムからいかにして要求 (解精度, 実行時間) に見合った並列メタ戦略を

設計するか、その方法論を提案、議論する。このため、まずメタ戦略の「粒度」を並列化の視点から定義する。本設計手法はこの粒度の観点から対象とするメタ戦略の性能を実験的に解析することにより、性能見積もり付の並列メタ戦略を設計する。

2 メタ戦略

工学分野を始めとする様々な分野において現れる問題の多くは現実的な時間で厳密な最適解を求めることが難しいとされる NP 困難のクラスに属していることが知られている。このためこれらの問題に対しては最適解ではないまでも比較的良い解を現実的な時間で求める (発見的) 近似アルゴリズムの開発がなされてきた。この目的を実現する一般的な枠組を提供するのがメタ戦略である [3]。すなわちメタ戦略とはある特定のアルゴリズムを指すのではなく、様々なアルゴリズムの総称、もしくは解探索の方針のことを指す。代表的なメタ戦略としては、遺伝アルゴリズム、アニーリング法、タブー探索法などが挙げられる。

本稿ではメタ戦略を局所探索法の一般化と捉えることにより、特定のメタ戦略だけではなく、多くのメタ戦略に適用可能な並列化メタ戦略アルゴリズム設計の方針を与えることを目指す。

2.1 局所探索法

局所探索法はある解 σ に少しの変形を加えることで得られる解の集合 $N(\sigma)$ (近傍と呼ばれる) 内を探索し、評価関数 f の下で改善解を発見するとその改善解の近傍から同様に探索を行い、近傍内に改善解が見つからなくなるまでこれらの操作を繰り返す手法である。局所探索法のアルゴリズムを次に示す。

アルゴリズム 局所探索法 $LS(N, f, \sigma)$

- (1): $f(\sigma') < f(\sigma)$ を満たす解 $\sigma' \in N(\sigma)$ が見つければ、 $\sigma := \sigma'$ とし、この操作を繰り返す。
- (2): σ を出力し終了。

2.2 メタ戦略アルゴリズム

ここでは多くのメタ戦略を捉えることのできる枠組みでアルゴリズムを記述する。近傍集合 Π 、初期解 σ 、評価関数 f 、目的関数 g 、解生成戦略 S 、初期探索パラメータ α_0 、初期状態 q_{init} 、終了状態 q_{halt} 、状態遷移関数 δ が与えられたときの一般的なメタ戦略のアルゴリズムは次の通りである。

アルゴリズム メタ戦略アルゴリズム

$MH(\Pi, \sigma, f, g, S, \alpha_0, q_{init}, q_{halt}, \delta)$

- (1): 暫定解 $\sigma_{best} := \sigma$ 、状態 $q := q_{init}$ 、探索パラメータ $\alpha := \alpha_0$ とする。
- (2): $\sigma^* := \sigma$ として解のコピーを行う。
- (3): 全ての近傍 $N_i \in \Pi$ に対して改善解が発見できなくなるまで $\sigma' := LS(N_i, f, \sigma)$ 、 $\sigma := \sigma'$ として局所探索を行う。局所探索中に $g(\sigma'_{best}) < g(\sigma_{best})$ となる解 σ'_{best} を見つけたら、 $\sigma_{best} := \sigma'_{best}$ とする。

- (4): 探索内容よりパラメータ α を更新する.
- (5): $f(\sigma^*) > f(\sigma')$ ならば $\sigma := \sigma'$, $q := \delta(q, \sigma, \alpha)$ として次状態へ.
- (6): $q = q_{halt}$ ならば σ_{best} を出力して終了.
- (7): 解生成戦略 $S(q)$ で解 σ を生成し (2) へ.

2.3 粒度

次にメタ戦略の「粒度」を定義する. メタ戦略の探索は次のように分類できる.

- シーケンス: 初期状態 q_{init} ~ 終了状態 q_{halt}
- フェーズ : 状態 q ~ 次状態 $q := \delta(q, \sigma, \alpha)$
- トライアル: 全ての $N \in \Pi$ に対する局所探索
 $LS(N, f, \sigma)$ が終了するまで
- サーチ : ある $N \in \Pi$ に対する局所探索
 $LS(N, f, \sigma)$ が終了するまで

それぞれの探索単位に対する粒度を, ここではシーケンス粒度, フェーズ粒度, トライアル粒度, サーチ粒度と呼ぶ. これらの粒度を示した図が図1である. 粒度の定義より2つの並列化手法が考えられる. 一つ目は各計算機にそれぞれ異なる初期解を与えて並列に独立なメタ戦略を行うシーケンス並列化である. 二つ目はフェーズ中のトライアルを各計算機で分担するトライアル並列化である. サーチ粒度での並列化も考えられるが, 一般には対象が非常に細かくなり膨大な通信回数になることが予想されるため今回は扱わない. また, フェーズ粒度については, 状態遷移が一本道になっており並列化の余地がないため扱わない.

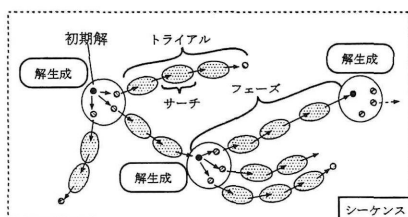


図 1: メタ戦略アルゴリズム

3 並列化

3.1 並列化のモデル

分散・並列計算のモデルとしては様々なものが提案されているが, 本稿では各プロセスが 1) 共有メモリを持たない, 2) メッセージ通信を行なうことができる, 3) 非同期である, といった条件を満たした環境の上での並列アルゴリズムの設計を考える. この条件に合うモデルとしては例えば

PC クラスタ [1] がある。PC クラスタは専用の並列計算機などと比べ安価で購入しやすいことから、様々な実用計算に利用されている。本稿で紹介する実験結果も (5 節参照), PC クラスタ上で実装されたアルゴリズムによるものである。

一般的に分散化のメリットとして、実行の高速化、得られる解精度の向上 (高精度化)、メモリの分散、負荷分散などが挙げられるが、本研究では特に高速化と高精度化の 2 点を目的とする。本分散環境では、各プロセス間での情報のやり取りにはメッセージ通信を用いる、という制約のため、通信には相応のコスト (通信時間) がかかる。また本研究では、マスタが各々のスレーブに対してプロセスを分配し、スレーブは割当てられたプロセスを実行するというマスタ・スレーブ方式を用いる。

3.2 シーケンス並列化

シーケンス並列化は探索開始から終了までのシーケンスを一つの粒度として扱った並列化手法である。図 2 はシーケンス並列化のイメージ図である。マスタは各スレーブにそれぞれランダムに生成した初期解を送り、スレーブからの暫定解の受信を全スレーブがそれぞれメタ戦略を終えるまで行う。スレーブはマスタから受け取った解からメタ戦略を終了状態まで行い、探索中に暫定解を発見した時点でマスタに逐一暫定解を送信する。シーケンス並列化では、複数回のメタ戦略が同時に行われるためスレーブ台数を増やせば増やすほど良い解が見つかり易くなる。しかし、各計算機によるメタ戦略の動作と 1 台でのメタ戦略のそれは異ならないため、本来のメタ戦略で発見できる解よりも良い解は望めない。スレーブの集合を P として、シーケンス並列化のアルゴリズムは次の通りである。

アルゴリズム シーケンス並列化-マスタ

$SP_{master}(\Pi, f, g, S, \alpha, q_{init}, q_{halt}, \delta, P)$

- (1): スレーブを $|P|$ 台起動する。
- (2): 各スレーブにそれぞれランダムに生成した解を送信し、 $g(\sigma_{best}) = +\infty$ とする。
- (3): メッセージ M_{halt} をスレーブ台数分受信するまで、スレーブから解 σ'_{best} を受信して $g(\sigma'_{best}) < g(\sigma_{best})$ ならば、 $\sigma_{best} := \sigma'_{best}$ とすることを繰り返す。
- (4): σ_{best} を出力して全スレーブを停止する。

アルゴリズム シーケンス並列化-スレーブ

$SP_{slave}(\Pi, f, g, S, \alpha, q_{init}, q_{halt}, \delta)$

- (1): マスタから初期解 σ を受信する。
- (2): $MH(\Pi, \sigma, f, g, S, \alpha, q_{init}, q_{halt}, \delta)$ を行い、探索中に新しい暫定解 σ'_{best} を見つける度にマスタに σ'_{best} を送信。
- (3): 終了メッセージ M_{halt} をマスタに送信して終了。

シーケンス並列化でのマスタ・スレーブ間の通信回数は、マスタが初期解を送る時のスレーブ台数回と、スレーブからマスタに解を送る時のスレーブ台数 \times 暫定解更新回 (一般にはこれは小さい) であると考えられるので、シーケンス並列化での通信コストは非常に小さい。また、マスタの負担が少ないため、台数をかなりの数まで増やすことができる。

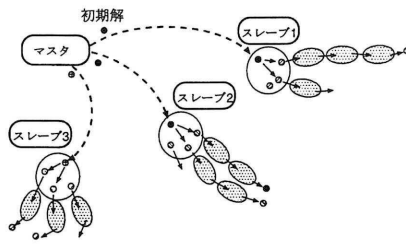


図 2: シーケンス並列化

3.3 トライアル並列化

通常メタ戦略は各フェーズにおけるトライアルを複数回行う。この性質を利用し、トライアルを複数台の計算機で同時に行うトライアル並列化により高速化が実現できる。また、各フェーズで少なくともスレーブ台数分のトライアルを行うことになり、より広い領域を探索することができるので、元のメタ戦略以上の解精度を得る可能性も考えられる。図3にトライアル並列化のイメージ図を載せる。トライアル並列化ではマスターの持つ解 σ と状態 q から解生成を行い、生成した解をスレーブに送信する。マスターはスレーブから解 σ' やその他の探索情報を受け取り、 $f(\sigma') < f(\sigma)$ ならば、次状態に遷移させる。そうでないなら状態 q のままで解生成を行い、スレーブに返すことを終了状態まで行う。スレーブはマスターから受け取った解をもとに、トライアルを行い、途中で発見した暫定解や最終的に得られた解をマスターに送信する。トライアル並列化では、1回のトライアルに対して2回の通信が必要になるため、探索中のトライアル数が多い場合には通信コストは比較的大きくなる。トライアル並列化のアルゴリズムは次の通りである。トライアルの部分を実行する部分を $Trial(\Pi, \sigma, f, g, \alpha)$ と表わすことにする。

アルゴリズム トライアル並列化-マスター

$TP_{master}(\Pi, \sigma, f, g, S, \alpha, q_{init}, q_{halt}, \delta, P)$

- (1): $\sigma_{best} := \sigma, q := q_{init}, \alpha := \alpha_0$ とする。
- (2): スレーブ台数分の解 $\sigma := S(q)$ を生成し、解とパラメータ $(\sigma, \sigma_{best}, \alpha)$ を各スレーブに送信する。
- (3): スレーブ $p \in P$ から解 $(\sigma', \sigma'_{best})$ を受信し、 $g(\sigma'_{best}) < g(\sigma_{best})$ ならば $\sigma_{best} := \sigma'_{best}$ とする。
- (4): 探索内容よりパラメータ α を更新する。
- (5): $f(\sigma) > f(\sigma')$ であるなら、 $\sigma := \sigma', q = \delta(q, \sigma, \alpha)$ として次状態へ。
- (6): $q = q_{halt}$ ならば、 σ_{best} を出力して終了。
- (7): 解生成戦略 $S(q)$ より解 σ を生成し、スレーブ p に $(\sigma, \sigma_{best}, \alpha)$ を送信し (5) へ。

アルゴリズム トライアル並列化-スレーブ

$TP_{slave}(\Pi, f, g)$

- (1): マスターから解とパラメータ $(\sigma, \sigma_{best}, \alpha)$ を受け取る。
- (2): $\sigma := Trial(\Pi, \sigma, f, g, \alpha)$ を行い、探索中 $g(\sigma'_{best}) < g(\sigma_{best})$ となる解 σ'_{best} を見つけたら、 $\sigma_{best} := \sigma'_{best}$ とする。
- (3): マスターに解 (σ, σ_{best}) を送信。

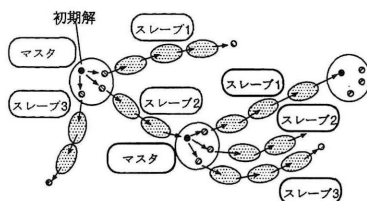


図 3: トライアル並列化

4 並列メタ戦略の見積もり

4.1 メタ戦略の性能見積もり

4.1.1 暫定解の収束解析

メタ戦略では探索が終了することがないため、終了条件を指定する必要がある。終了条件を実行時間とすることが多いが、実行時間と解精度はトレードオフの関係を持つため、単純に解精度による比較、実行時間による比較ができない。このためここでは、どのような解をどの程度の時間で見つけることができるかでメタ戦略の性能を評価することを考える。

最小化問題に対するメタ戦略ならば、最適解の下界(発見されていれば最適解自身)にできるだけ近い解を、できるだけ高速に見つける性能が要求される。そこで並列化の対象とするメタ戦略の性能を、探索中に発見される暫定解とその解を得た時間により評価する。図4はある問題に対するメタ戦略を500秒でそれぞれ初期解を変えて10回実行し、横軸を時間(秒)、縦軸を解の目的関数値と下界との誤差として実験結果を表わしたものである。この例ではいずれの探索も探索開始から数十秒で急速に暫定解を発見し、下界の1.001倍の目的関数値を持つ解に辿り着き、300秒以降は新しい暫定解の発見が跡絶えているのが分かる。この最終的に得られた暫定解を収束解と呼び、収束解を発見した時間を収束時間と呼ぶ。一方、初期解により収束解とその収束時間にもばらつきが見られるため、暫定解発見の平均的な振る舞いで評価することを考える。各時間における暫定解10個分の平均を探索中の各時間において求め、それぞれの暫定解の平均値により、平均収束解とその収束時間が得られる。

4.1.2 暫定解の分布解析

あるメタ戦略の暫定解の収束を調べ、最終的に得られる解にばらつきが見られた場合、その暫定解の分布を調べることで対象とするメタ戦略の性能を見積もることを考える。これは最終的に得られた解を標本とし、その標本分布から母集団の分布を推定する統計的推定により求める。ここで母集団は実行可能解の集合であり、その母数や母集団の分布を知ることは難しいため、推定は点推定という手法で行う。点推定は、母集団を正規母集団と考えて標本平均と標本分散から正規母集団を推定する手法である。ここで、 $seed$ を変えて k 回の実行を行い、得られた暫定解が目的関数値 x_i ($i = 1, 2, \dots, k$)を持つ場合、

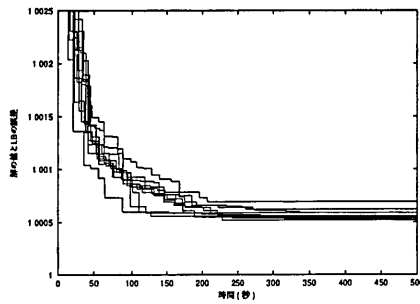


図 4: 暫定解の収束

$$\text{標本平均} \quad \bar{x} = \frac{1}{k} \sum_{i=1}^k x_i$$

$$\text{標本分散} \quad s^2 = \frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^2$$

と表わすことができ、この標本平均 \bar{x} と標本分散 s^2 をそれぞれ平均、分散とする正規分布 $N(\bar{x}, s^2)$ を推定母集団の分布とする。図 5 の実線は図 4 で最終的に得られた暫定解の分布を、横軸に暫定解の目的関数値と下界との誤差、縦軸に暫定解の得られた割合として表わした図である。また、図 5 の点線は実線の暫定解の分布を標本として点推定を行い、得られた推定母分布である。この推定分布を発見が期待される分布として並列メタ戦略の性能評価に用いる。

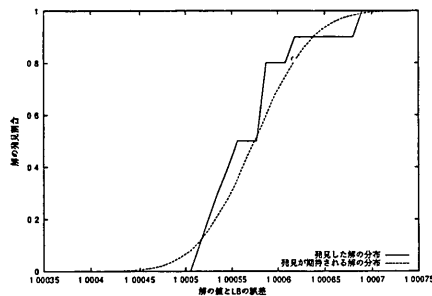


図 5: 暫定解の分布と発見が期待される解分布

4.2 シーケンス並列化の見積もり

対象とするメタ戦略から得られた暫定解の収束解析、分布解析により、シーケンス並列化の性能見積もりを行う。収束解析で得られたデータからある時間 t における暫定解の分布が得られる。各

時間 t での分布を $F_t(x)$ とすると、2 台でのシーケンス並列化で期待される解の分布 $F_t^{(2)}(x)$ はそれぞれ独立な解の確率変数 X_1, X_2 を用いて $F_t^{(2)}(x) = P(\min\{X_1, X_2\} \leq x)$ である。これは *seed* を変えたメタ戦略の動きが独立であるとする $1 - (1 - F_t(x))^2$ に等しいので、 n 台でのシーケンス並列化の性能見積もり $F_t^{(n)}(x)$ は $1 - (1 - F_t(x))^n$ で表わすことができる。

4.3 トライアル並列化の見積もり

4.3.1 並列度解析

トライアル並列化を行う場合、各フェーズにおけるトライアル数を調べることから始める。各フェーズにおけるトライアル数が使用可能計算機数よりも多い場合、効率の良い高速化を望むことができる。また、各フェーズにおけるトライアル数が少ない場合には、各フェーズで少なくとも使用台数分のトライアルを行うことになり探索領域が広がるため、未発見の解を見つける可能性が増えることが考えられる。以上から、並列度解析を行うことで、高速化と高精度化の並列化の方針を決めることができる。ここで、高精度化は同じ終了時間内に高精度な解を発見することができることを意味するので、対象とするメタ戦略での収束解を早い時間で見つけることのできる高速化の意味も持つ。図6はある問題に対するメタ戦略について、各フェーズにおけるトライアル数を表わしたものである。探索開始のフェーズのインデックスを1、として順に各フェーズ毎のトライアル数を縦軸に取っている。図6より、トライアル数は多くて150程度で、平均は約4であり、トライアル数が1のフェーズが7割を占める。これより、このメタ戦略には台数幅解析を行い、高精度化の影響を調べる必要があると言える。

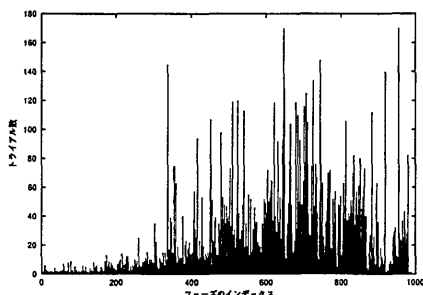


図 6: 各フェーズにおけるトライアル数

4.3.2 台数幅解析

台数幅解析では n 台でのトライアル並列化の影響を調べるために、各フェーズでのトライアルを少なくとも n 台行うことで、トライアル並列化の性能をシミュレートする。図7はある問題に対するメタ戦略に各フェーズのトライアルを最低10回は行うという制約を加えたもので、元のメタ戦略よりも収束に時間がかかるが、元のメタ戦略よりも良い解に辿り着いている。このことはトライアル並列化による高精度化が可能であることを意味する。図6の例では各フェーズでのト

ライアル数が1であるものが多いため、少なくともトライアルを10回以上行う場合、本来のメタ戦略に要した時間の最大10倍の時間で実験を行ったものが、並列化のより正確な見積もりになると考えられる。

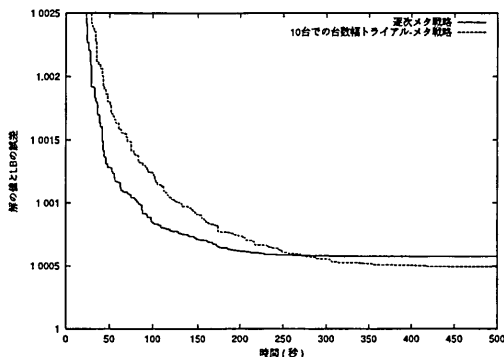


図 7: 逐次メタ戦略と10台での台数幅トライアル-メタ戦略による解の収束の平均

4.3.3 パラメータ更新頻度解析

トライアル並列化をする際に問題となるのは、探索パラメータをどのように扱うかである。探索パラメータの更新されるタイミングによっては元のメタ戦略に忠実なパラメータ更新ができない場合がある。元のメタ戦略で探索パラメータの更新がフェーズ毎に行われる場合、マスターはパラメータが更新される度に各スレーブに送信すれば良い。しかしパラメータの更新頻度がトライアル毎である場合は単純ではない。本来のメタ戦略で i 番目のトライアル t_i でのパラメータを α_i とすると、 α_{i+1} は t_i の探索内容と α_i によって調節が行われる。これに対してトライアル並列化はパラメータの調整に次の2手法を採用している。 k 台での並列化の場合、一つはトライアル $t_i, t_{i+1}, \dots, t_{i+k}$ を同時に同じパラメータ α_i で行う手法で1つのフェーズは固定。また、パラメータを重要視する並列化では、トライアルが終わる度にマスターがそのトライアルの内容からパラメータを更新し全てのスレーブにブロードキャストする方法も考えられる。これらのパラメータ更新のタイミングが異なる動作をシミュレートするのがパラメータ更新頻度解析である。本解析では、元のメタ戦略に手を加え、パラメータの更新タイミングを(1)フェーズ粒度、(2)サーチ粒度に変更したものをそれぞれ実装して解の収束と分布を調べる。トライアル並列化はこの解析の結果性能の良かったタイミングで実装を行う。

4.4 解析からの見積もり

以上の3つの解析結果からトライアル並列化の見積もりを行う。まず(1)パラメータ更新頻度解析により決定した粒度でパラメータの調節を行い、(2)各フェーズで使用台数分以上のトライアルを行う台数幅解析を組み込んだメタ戦略で実験を行う。さらに(3)高速化の見積もりを行った結果得られたのが、図8である。高速化は、元のメタ戦略の並列度解析で得られた各フェーズ毎

のトライアル数を台数分で畳み込み、高速化の割合を算出したものに、総トライアル数×1回の送受信にかかる時間を組み込むことで見積もる。

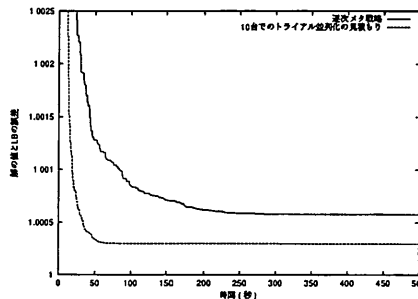


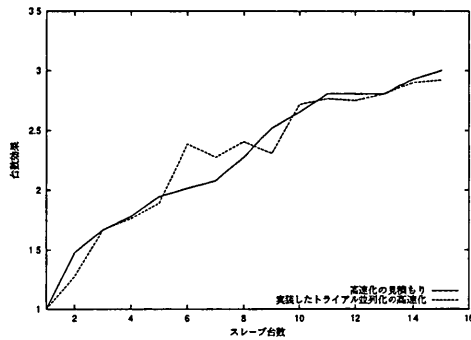
図 8: 収束解析によるデータの平均と 10 台でのトライアル並列化の解の収束見積り

5 実装と実験結果

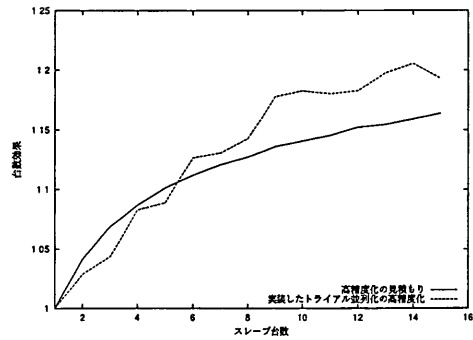
本実験では、一般化割当問題 [4] に対するタブー EC 法 [2] を対象としてその解析からそれぞれの手法での並列タブー EC 法の見積もりを行い、実装した結果との比較を行う。一般化割当問題は代表的な組合せ最適化問題の一つであり、ベンチマークとして様々な種類のインスタンスが公開されている (<http://www-or.amp.i.kyoto-u.ac.jp/members/yagiura/gap/>)。実験にはこのうち、d401600 という比較的大規模なサイズのインスタンスを使用する。実験環境は CPU: Pentium4 2.26GHz, MEM: 512MB, OS: Debian2.20, コンパイラ: gcc2.95.4, 通信ライブラリ: pvm3.4[1] である。使用計算機台数は 2~16 台でそれぞれにおいてシーケンス並列化、トライアル並列化の見積もりと実装して得た実験結果から、その高速化と高精度化の比較を行う。実験はそれぞれ 1 回 500 秒で *seed* を変えて 10 回ずつ行う。高速化はメタ戦略の平均収束解 (LB の 1.000575 倍の目的関数値を持つ解) を平均収束時間 (337 秒) の何倍早い時間で見つけることができるか、で評価する。高精度化は平均収束解と LB の誤差 (0.000575) を 1 とし、何倍 LB に近づいたかで評価を行う。図 9(a), 図 9(b) はそれぞれシーケンス並列化と見積もりの高速化、高精度化の比較、図 10(a), 図 10(b) はそれぞれトライアル並列化と見積もりの高速化、高精度化の比較である。図は横軸にスレーブ台数、縦軸に台数効果を取ったものである。高速化はシーケンス並列化、トライアル並列化共に良い見積もりが得られていることがわかる。一方、高精度化についても台数が増えるにつれて若干誤差が大きくなっているものの、両者とも見積もり値は実装値に比較的近い値となっている。

6 おわりに

今回、一般のメタ戦略に適用できる「粒度」の枠組みと、粒度に基づいた二つの並列化手法の実験的解析による性能見積もり方法を提案した。従来のメタ戦略アルゴリズム並列化では、対象となるアルゴリズムの動作を詳細まで知っておく必要があったのに対し、本設計手法では、対象メ

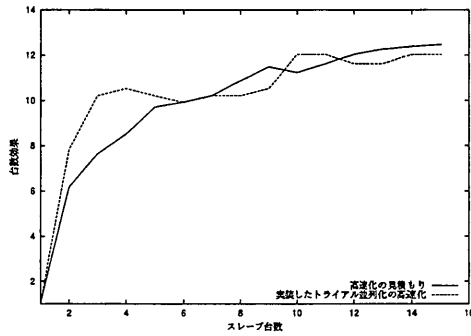


(a) (高速化による比較)

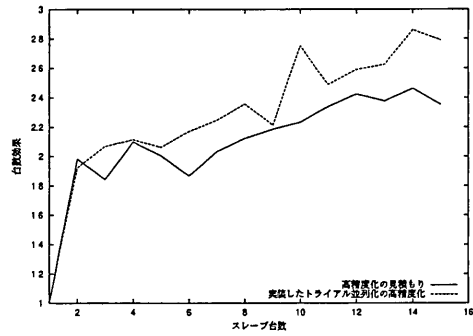


(b) (高精度化による比較)

図 9: シーケンス並列化の見積もりと実装値



(a) 高速化による比較



(b) 高精度化による比較

図 10: トライアル並列化の見積もりと実装値

タ戦略の並列性を粒度にしたがって実験的に解析するだけで、(提案した見積り方法の結果に基づき) その並列性を引き出すような並列メタ戦略アルゴリズムを設計することが可能となる。

本提案手法に基づく一般化割当問題に対するタブー EC 法の並列化例では、計算実験の結果、提案した性能見積もりが実際の実験値に近い値であることが確認された。また、紙面の都合上詳しく述べることはできなかったが、これらの結果では、いくつかのベンチマーク問題に対する最良解を更新している。今後は、ある問題に対する専用並列メタ戦略アルゴリズムと本並列化手法に基づく並列アルゴリズムでの性能比較を行う予定である。

参考文献

- [1] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam. PVM: parallel virtual machine - a users' guide and tutorial for networked parallel computing,

MIT Press, 1994.

- [2] M. Yagiura, T. Ibaraki and F. Glover. An Ejection Chain Approach for the Generalized Assignment Problem, *INFORMS Journal on Computing*, to appear.
- [3] M. Yagiura and T. Ibaraki. On metaheuristic algorithms for combinatorial optimization problems, *Systems and Computers in Japan*, 32, Issue 3, 33-55, 2001.
- [4] S. Shani and T. Gonzalez. P-complete approximation problems, *J. ACM*, 23, 555-565, 1976.
- [5] V.-D. Cung, S. L. Martins, C. C. Ribeiro, and C. Roucairol. Strategies for the parallel implementation of metaheuristics, *Essays and Surveys in Metaheuristics*, 263-308, 2001.

有限要素法による大規模音場解析

大鶴 徹* 富来礼次** 岡本則子*** 福田みどり***

1 はじめに

世の中、「重厚長大」からの脱却が叫ばれて久しい。しかし「(超)大規模・(超)高精度・超並列」といったコピーに胸をときめかすのは「より速く・より高く・より強く」と刷りこまれた東京オリンピック世代のみではないと信じたい。

建築音場の予測や設計には長い間、Ray Methodと呼ばれる幾何音響(音を光のように直進するエネルギーとして近似的に取り扱う方法)に基づく技術が多用されてきた。初期には人手によるRay Methodを用いた作図を用いた検討が行なわれていたが、計算機による建築音場の音線追跡法等が研究開発され、ごく最近ではHagia Sophia内部における美しいコーランの響きまでもシミュレートされている¹⁾。実務に活用されるこの技術を"better than nothing"とまでは言わないが、幾何音響的なエネルギーの重畳のみでは音場のモードや回折等、波動が織り成す自然現象の科学的解明は難しからう。

建築音場の科学的解明には波動方程式を離散的に解く、即ち、差分法・有限要素法・境界積分方程式法(境界要素法)等の数値解析技術が必要となる。九州大学大型計算機センターでM780やM382が稼働していた80年代から、筆者らは建築音環境の解析のコア技術として有限要素法を用いてきた。日本を代表する電算センターにおいてすら一般ユーザーが使える主記憶が16や60 Mbyteに限られていた時代では、3次元建築空間内音場を有限要素法で解くという計画は世界の至る所で"funny joke"として受け止められた。

しかし、Gbyteのメモリの処理能力を有するPCが珍しくない昨今では、200m³程度の室内音場であれば誤差の少ない直接法で解くことができるし、数Gbyteのメモリを有するWS上で反復法を用いれば、1000席程度のコンサートホール内部の10000m³超の空間を1 kHz程度まで解析できる。

建築音場解析に用いられる数値解析技術の特徴は以下の2点ではなからうか：

- 1) 対象の大きさ、複雑さ
- 2) 人間聴覚との対応による解析結果の評価

境界は建築壁体であるから1)の克服にはCADの活用が有効だが、CADの活用は建築分野に限った話でもなく、目新しくもない。但し形状を含め雑多な境界の吸音性状の的確な数値モデル化は興味ある課題として重視できよう。Hagia Sophiaのモザイクタイルのインピーダンスをいかにして与えれば良いのか。これに答える手がかりは文献2)等にある。

数値解析結果の精度の評価を行う場合、建築音響分野ではオーロライゼーション(可聴化)がキーテクノロジーの一つとされる。ヒトの聴覚による演算結果判断の厳しさは、0~120dB程度という可聴域の広さからも推察できる。Sabineは自らの耳と自作機器を用いた実験を重ね、60 dBの音響エネルギー減衰(1/10⁶倍)に要する時間を以て「残響時間」を定義した。このことから、建築音響的観点からは、fff(フォルテッシモ)からppp(ピアノッシモ)あるいは曲間の静寂までを網羅するには60dBのダイナミックレンジを要する、と推量される。これら聴覚の120ないし60dBというダイナミックレンジは、音圧としては10⁻⁶若しくは10⁻³という数値計算精度の要求水準につながる。

ビジュアルライゼーションも現象理解の手段として有効であり建築音場解析の分野でも活用されている。但し、ダイナミックレンジの観点からはヒトの聴覚は視覚に比べ遥かに鋭敏で、その鋭敏な聴覚での検証プロセスを前提としている点こそ、建築音場解析の特徴と言えよう。

2 有限要素による音場解析

2.1 離散化方程式

音場について構成した汎関数へ変分原理を適用し、周波数 ω の定常問題と仮定すれば次の離散化方程式が得られる³⁾。

所属：* 大分大学工学部福祉環境工学科建築コース

** 大分大学VBL博士研究院

*** 大分大学大学院工学研究科福祉環境工学専攻

otsuru@cc.oita-u.ac.jp

reijtom@arch.oita-u.ac.jp

$$([K] + i\omega[C] - \omega^2[M])\{p\} = -i\omega\rho v_0\{W\} \quad (1)$$

なお、 ρ は空気密度、 v_0 は外力の振動速度、 $\{p\}$ は音圧ベクトル、 $\{W\}$ は配分ベクトル、 $[K]$ 、 $[M]$ はそれぞれ音響系の運動とポテンシャルエネルギーに対応する行列で、詳細な定義は文献4)を参照されたい。また $[C]$ は次節に示す減衰行列である。

2.2 音響要素

一般に音響に関する3次元有限要素解析では、アイソパラメトリック8節点要素³⁾⁵⁾や20節点要素³⁾が知られている。筆者らはNatural Spline関数を用いたアイソパラメトリック27節点音響要素(Spl27)を開発し、上記の要素との比較を種々行いその有効性を確認した⁶⁾。Fig.1は、固有値解析の近似精度を分割数との関係を示したものである。分割数を増やしていく際の近似誤差の減少が、Spl27は他の要素に比べ速く安定している。

建築音場解析で重要な境界の吸音は、表面インピーダンスモデルと吸音要素モデルの2者を用いている。前者のモデルは、局所作用の仮定が成立する場合に、要素減衰マトリクス $[C]_e$ は壁面のノーマルインピーダンス比を z_n として、

$$[C]_e = \frac{1}{c} \iint_e \frac{1}{z_n} \{N\}^T \{N\} dx dy \quad (2)$$

で定義できる。ここに、 c は音速、 $\{N\}$ は内挿関数である。またサフィックス e は要素に関する行列や積分であることを示す。一方、局所作用の仮定が成立しない場合、例えば多孔質材のRayleighモデルに基づく減衰行列はCraggsの式を改良した次式で与えられている⁷⁾。

$$[M_{ab}]_e = \epsilon K_s \Omega_{ab} [M]_e, \quad [C_{ab}]_e = \epsilon \frac{R \Omega_{ab}}{\rho} [M]_e, \quad [K_{ab}]_e = \epsilon [K]_e \quad (3)$$

但し、

$$\epsilon = \frac{i\omega\rho}{(R + i\omega\rho_{ab})} \quad (4)$$

なお、 K_s 、 Ω_{ab} 、 R および ρ_{ab} はそれぞれ材の構造係数、ポロシティ、流れ抵抗率および密度である。実際の解析にあたってはこれら(2)、(3)の各式の減衰モデルを適宜使い分けるが、与える結果の相違については3章で垣間見て頂きたい。

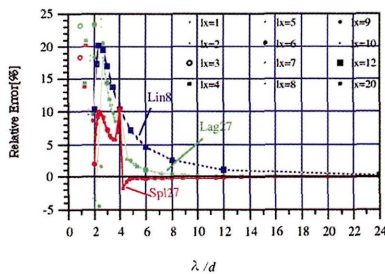


Fig.1 各種音響要素を用いた場合の要素分割(λ/d)と固有振動数近似の相対誤差⁶⁾

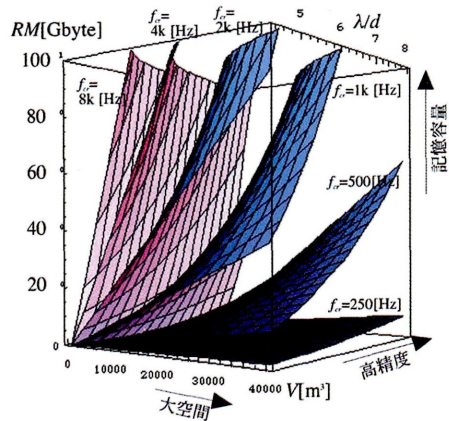


Fig.2 Spl27を用いた場合の $V[m^3]$ 、 λ/d 、 $RM[Gbyte]$ の関係⁷⁾

2.3 有限要素法による建築音場解析の数値解析的な特徴

簡単のため(1)式を次のように記すこととする。

$$[A]\{p\} = \{f\} \quad (5)$$

室内の音圧分布を求めるには、この線形方程式を解けばよい。但し $[A]$ は複素対称疎行列（帯行列でもある）となる。この方程式の実数化は容易だが係数行列は規則性の高い実非対称疎行列となる。

よく知られているように線形方程式の解法に直接法と反復法がある。前者による解析では帯行列の要素の格納が必要で、記憶容量の制約のため3次元（音場）解析は容易ではない。これに対し後者を用いれば、各行あたりの非ゼロ要素数 $\times D.O.F.$ （総自由度数）のみを記憶させればよく、3次元解析も十分可能となる。

2.2節の27節点要素を用いた場合、行あたりの非ゼロ要素数は125、対称性を利用すればさらに必要記憶容量を減じ得る。Fig.2はSpl27を用いた場合の、室容積 $V[m^3]$ 、波長/節点間距離 λ/d 、必要記憶容量 $RM[Gbyte]$ をそれぞれ3軸にとり解析上限周波数 $f_{cr}[Hz]$ をパラメータとした説明図である。なお RM は余裕を見込んだ筆者らの概算値だが、現状のスーパーコンピュータの記憶容量を考えた場合、巨大なコンサートホール内部音場の自在な解析も夢物語ではなくなっているようである。

3 解析例

3.1 実在音場解析例(1)：残響室解析⁷⁸⁾

まず、境界条件が比較的単純な室として残響室を対象に有限要素解析を行い、実測音圧分布と比較した例を紹介する。Fig.3は大分大学総合情報センター地階大残響室の概要とメッシュ分割例である。この室はわが国で標準とされてきた不整形残響室($V=168m^3$)で、音の拡散性を高めるため平面形状は五角形、各壁面や床面と天井面は互いに平行にならないよう配慮されている。各壁面は反射率の高いコンクリート製で、実測用音源スピーカも壁面に埋め込み可能な限り凹凸を減じた。また吸音の影響を検討する目的で、室の一角にグラスウールを設置している。

有限要素解析にはSpl27を用い、500Hz程度を解析の上限と想定した上で総自由度数 $D.O.F.$ は16767とした。この程度の規模の解析であれば連立一次方程式の解法に直接法を用いることができる。ここでは周波数 ω 毎に修正Cholesky法を適用した。結果として16767点における複素音圧ベクトル $\{p\}$ を得る。

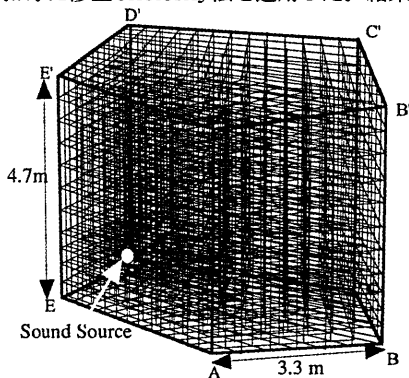


Fig.3 不整形残響室の概要とメッシュ分割例

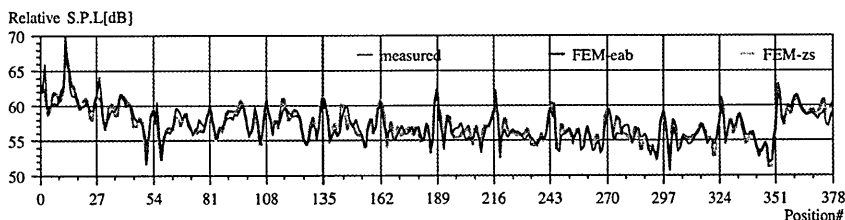


Fig.4 残響室内378点における音圧レベルの有限要素解析値と実験値の比較⁸⁾

なお壁面の表面インピーダンスは、実測残響時間から吸音率を求め、この値から虚数部を0としたインピーダンスを与えた。グラスウールについては2.2節の2つのモデルを用い比較した。なお、流れ抵抗係数その他は、管内法で測定した同一材のノーマルインピーダンスとMikiの実験近似モデル⁹⁾をもとに推定した値とした。

一方、音響の実測にはマイクロフォンとリアルタイムアナライザ等を用いるが、如何に「多チャンネル」と謳ってはいても一般には2~4ch、相当に高額な機器でもせいぜい100chというのが建築音響計測の現状である。また純音や狭帯域の音の測定では室内に生起するモードの影響が大きく、マイクロフォン位置の設定には細心の注意を要する。筆者らの研究室でも小規模ではあるが2次元マイクロフォン移動ロボットを試作し一部の実験では用いているものの、数十m²の領域を覆う不規則なメッシュ上の音圧を精密かつ高速に測定するこの実験では高い意識を備えた人間の能力の方が今のところ勝っている。特に壁際の音圧測定にはマイクロフォンの方向や位置の設定に細心の配慮が不可欠となる。従って本研究の推進にあたっては、建築音響学の基礎と高い目的意識を備えた卒論生の力に負うところが大きい。こうした実験上の制約のため実測値と有限要素解析結果の比較は床から1.2m高さの平面上の378点としている。また位置や周波数の変動に対し鋭敏な純音ではなく1/3オクターブ帯域での検討とした。

Fig.4に中心周波数200 Hzの1/3オクターブバンドに関する実測値(記号: measured)と、(2)式の吸音モデルを用いた有限要素解析値(同: FEM-eab)、(3)式の吸音モデルによる値(同: FEM-zs)を比較して示す。壁際の音圧上昇やピークディップの変動まで含め全般的な3者の一致は良好である。また両吸音モデルのいずれを用いても実測値を基準とした378点の音圧レベルの平均残差は1.0dB以下であった。なお用いる吸音モデルによる相違はこの例では顕著ではない。しかし、材を設置する位置が壁でなく背後に壁がない場合などでは吸音材の近傍で差が大きくなること、その場合は(3)式によるモデルの方が実測に近似の値を与えること、等を、別途、詳細な検討で確認している。

3.2 実在音場解析例(2): コンサートホール¹⁰⁾

大規模解析の一例として、大分市内の音楽用中ホール(Fig.5、約700席)の有限要素解析の概要を紹介する。ここでは基本的な可能性の検討として、ステージとホールを包含する約12000m³の空間を解析対象に設定し、床や壁等には相応の複素インピーダンスを与え(2)式によるモデリングを行った。またSpl27を用い上限周波数を500Hz程度として、複素D.O.F.を約3700000に設定した。連立一次方程式の解法にはBiCG法を適用したが、予め前節の残響室等同一の対象をBiCG法と直接法の双方で解析し、得られる結果間の誤差が10⁻⁶以下となることを確認している。

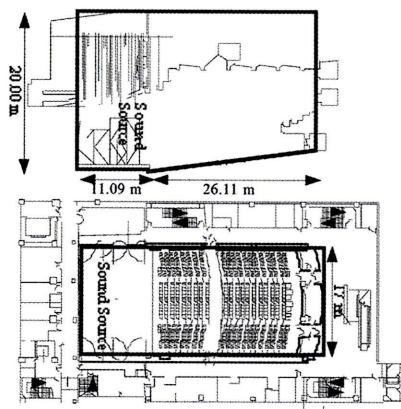


Fig.5 解析対象の音楽ホール断面図と平面図¹⁰⁾

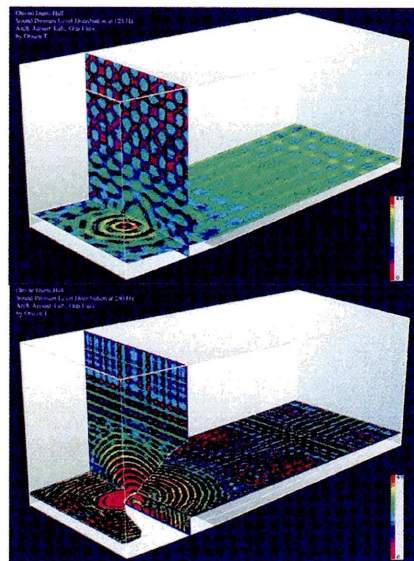


Fig.6 音楽ホール内部音圧レベル分布の有限要素解析結果(上: 125Hz, 下: 250Hz)¹⁰⁾

結果の一例として125Hzと250 Hzの室内音圧レベル分布をFig.6に示す。これらの周波数ではモードの影響が大きいことや周波数が高くなれば拡散性が高まる傾向が表れており物理的考察と一致する。またこれらの解析はGS320で約5 G Byteのメモリーと約6時間の演算時間を要したが、前述のオーロライザー等を実現するには演算時間の短縮が不可欠となる。そのためには、まず、前処理を含めた反復法の高速化が考えられる。併せて、並列機の効率的利用も不可欠である。

なおここで紹介した解析手法では周波数領域での応答を個々の ω 毎に対角スケーリングのみ施した連立方程式より求めている。従って ω 毎にプロセッサを割当て並列化を図れば多数の周波数応答の同時算定は容易で、また筆者らも既に九州大学大型計算機センターへVPP700が導入された時から実行してきた。現在は、ICCG法の適用やBiCG法の改良あるいはこの分野に適した解法やサブルーチンライブラリの選択、さらにはセンターのGS320やVPP5000の活用等々、により実用的解析が可能となるよう検討を重ねている。

むすび

大規模数値解析など単純作業の繰り返しに過ぎず所詮は釈迦（基礎理論）の手の上だ、との批判もある。しかし神ならぬ（仏でもない）衆生は有限の時間を生き、ヒトは有限の音響エネルギー減少（ $-60\text{dB} = 1/10^6$ ）を以て音の「消滅」と意識する。ビザンチン様式のHagia Sophia はかつてはキリスト教会、その後モスクとなり、現在は博物館として1500年の時を生き続けている。その中に在る約260000 m^3 の空気粒子が織り成す音の生滅は様々な感覚や感情をヒトに生起してきたものと想像できる。その境界（建物）の活きる時間に比べあまりの瞬時に消え去る音を光(Ray Method = 幾何音響)ではなく音そのものとして解明するには、基礎理論の充実とともに数理解析技術と計算機の更なる発展に期待するものである。末筆ながら日頃よりご助言を頂いている九州大学情報基盤センターの藤野清次教授と南里豪志助教授に記して謝意を表します。

参考文献

- 1) J. H. Rindel, C. L. Christensen, "Room acoustic simulation and auralization -How close can we get to the real room?-", WESPAC VIII, Proceedings/CD-ROM (Melbourne, 2003)
- 2) Y. Takahashi, T. Otsuru, et.al., "An efficient impedance and absorption coefficient measurement technique for digital simulations by using environmental anonymous noise," Technical Acoustics Vol. 21, pp.263-266(2002)
- 3) 加川幸雄、「有限要素法による音響／振動解析」、培風館(1981)
- 4) 日本建築学会編、「室内音場予測手法-理論と応用-」、第3章2節、丸善(2001)
- 5) 大鶴徹、「建築音響解析ライブラリACOUSTの開発」、九州大学大型計算機センター広報Vol. 26, No.5、pp.549-554(1993)
- 6) T. Otsuru, R. Tomiku, "Basic characteristics and accuracy of acoustic element using spline function in finite element sound field analysis," J. Acoust. Soc. Jan. (E), Vol.21, No.2, pp.87-95 (2000)
- 7) T. Otsuru, et.al., "Accuracy of dissipation models in the finite element sound field analysis of rooms with absorptive materials," Inter-Noise 99, pp.1663-1666(1999)
- 8) 富来礼次、大鶴 徹、「有限要素法による不整形残響室内音場解析」、日本建築学会計画系論文集第551号、pp.9-15 (2002)
- 9) Y. Miki, "Acoustical Properties of Porous Materials -Generalizations of Empirical Models-", J. Acoust. Soc. Jan. (E), Vol. 11, No. 1, pp.25-30(1990)
- 10) T. Otsuru, et. al., "Large Scale Sound Field Analysis by Finite Element Method Applied onto Rooms with Temperature Distribution," Proc.of ICSV 9/CD-ROM (2002)

CVSでファイル管理(2)

池田 大輔*

1 はじめに

CVSとは Concurrent Versions System の略で、バージョン管理システムの一つです。前回 [5] の記事では、UNIX 上での CVS の基本的な使い方を紹介しました。これにより、CVS の管理対象となる複数のファイルのバージョンを管理し、任意の時点のファイルを取り出すことができるようになりました。さらに、枝を作り目的の異なる作業を独立に並行して行なうことができるようになりました。ただし、前回の記事では、レポジトリは手元の計算機にあると仮定していました。

本稿では、まずネットワーク上にある CVS レポジトリの利用方法を説明します。利用する計算機は UNIX で、SSH を使って安全に通信できるようにします。次に、Windows や Machintosh 上で GUI 付きの CVS クライアントを利用して、ネットワーク上の CVS レポジトリを利用する方法を説明します。ここでも、UNIX の場合と同様に SSH を用いて安全に通信します。最後に、CVS のちょっとした便利な使い方なども紹介したいと思います。

レポジトリをネットワーク上に置くことで、他人との共同作業が可能になります。また、ノートパソコンや自宅のパソコンからでも、同じファイルにアクセスできるようになります。Windows や Machintosh の場合は、UNIX に比べると手元の計算機に CVS サーバを置きにくいこともあるでしょう。このような場合にも、レポジトリがネットワーク上にあれば、ユーザは Windows や Machintosh には CVS クライアントをインストールするだけで済みます。

2 ネットワーク上のレポジトリ

CVS のデータ保管場所であるレポジトリは、ネットワーク上にある別のマシンにあっても構いません。この節では、UNIX 上の CVS クライアントからネットワーク上のレポジトリへのアクセス方法を説明します。次に、このようなレポジトリを提供するために必要なサーバの設定を説明します。また、複数の人が同じファイルを編集するときに必要な設定もあわせて説明します。

2.1 アクセス方法

レポジトリの指定は、レポジトリがネットワーク上にあるかないかにかかわらず、環境変数 `$CVSROOT` で指定するか、`cvs` コマンドの引数 `-d` により指定します。

*情報基盤センター研究部 <mailto:daisuke@cc.kyushu-u.ac.jp>

前回の復習になりますが、手元の計算機にレポジトリがある場合は単にディレクトリを指定します。環境変数で指定する場合は、例えば

```
% setenv CVSROOT $HOME/CVS
```

とします。引数で指定する場合は

```
% cvs -d $HOME/CVS checkout project
```

などとします。一度 checkout してしまえば、その後この作業用コピーで作業 (diff や commit など) する場合に -d によるレポジトリの指定は不要です。この仕組みについては 5.2 節で説明します。

ネットワーク上にレポジトリがある場合、主に rsh など他のプログラムを利用して接続する方法 (ext 接続) と、CVS がサーバとなり直接接続する方法 (pserver 接続) の二つの方法があります。

ext 接続では、外部プログラムとして ssh が利用できます。この場合、認証に用いるパスワードやその後の通信はすべて暗号化されます。pserver 接続は、デフォルトの設定ではパスワードや通信は暗号化されませんが、SSH のポート転送を利用することで暗号化することが可能です。以下で、それぞれについて説明します。

2.1.1 外部プログラムによる接続 (ext 接続)

ext 接続の場合、外部プログラムとして rsh や ssh が利用できますが、安全性の面から ssh による接続方法を説明します。ただし、レポジトリのある計算機 (これを CVS サーバと呼ぶことにします。) にユーザとして登録されていて、ログインが可能でないといけません。

どのプログラムを用いるかは、環境変数 \$CVS_RSH で以下のように設定します。

```
% setenv CVS_RSH ssh
```

レポジトリは単なるディレクトリではなく “:ext:user@server:/directory” となります。ここで user とは、レポジトリのある計算機に登録されたユーザ名です。手元の計算機とユーザ名が同じ場合はユーザ名と “@” を省略でき、“:ext:server:/directory” となります。

例えば、レポジトリを kyu-cc 上の /home/usr1/a70011a/CVS とする場合は

```
% setenv CVSROOT :ext:a70011a@kyu-cc:/home/usr1/a70011a/CVS
```

とします。

これで checkout などを実行します。SSH の公開鍵を CVS サーバに登録している場合はパスフレーズを、そうでない場合は通常のパスワードを要求されます。パスフレーズやパスワードは、cvs のサブコマンドを利用するたびに要求されます。一方、公開鍵を登録していて、さらに ssh-agent と ssh-add を利用している場合は、認証は自動的に行なわれ、明示的なパスフレーズやパスワードの要求はありません。

以後、この作業用コピーで作業をする場合は、「貼りついたタグ (Sticky tag) (5.2 節参照) の働きにより、レポジトリを明示的に指定する必要はありません。環境変数ではなく、cvs コマンドの引数でレポジトリを指定していた場合でも、以後この作業用コピーを利用する場合は、

レポジトリの指定は不要です。

2.1.2 CVSによる直接接続 (pserver 接続)

ext 接続とは異なり、pserver 接続では、利用者は CVS サーバのアカウントは必要はありません。この計算機にログインができなくても、別途 CVS 用のユーザ登録をすればレポジトリが利用できます。

pserver 接続の場合は CVS をサーバとして動かす必要があります。また、ユーザ登録が別途必要です。登録は、レポジトリのある計算機に登録されていても、別途必要です。これらの設定方法については 2.3 節を参照してください。

pserver 接続では、通信が暗号化がされるわけではありません。認証に利用するパスワードもそのままネットワークを流れます。これを安全に利用するには SSH のポート転送を利用します (2.2 節)。

レポジトリは前節と同様、サーバ名とそのディレクトリを指定しますが、最初は “:ext:” ではなく “:pserver:” となります。つまり “:pserver:user@server:/directory” となります。この場合の *user* は CVS 用に登録したユーザ名です。

pserver 接続の場合は、最初に以下のようにログインする必要があります。

```
% cvs -d :pserver:user@server:/directory login
```

パスワードを要求されますので、入力してください。

認証が成功すれば、入力したパスワードは \$HOME/.cvspass に暗号化をして記録されます¹。そのため、一度 login してしまえば、その後 checkout や update などする場合にパスワードは不要です。

2.2 pserver 接続のポート転送

pserver 接続では、CVS サーバとの間の通信は暗号化されません。しかし、SSH のポート転送機能を利用すればより安全に通信ができます。

SSH のポート転送を利用する場合は、CVS を利用する前に ssh でログインしておきます。

```
% ssh -N -L localhost:pserver:2401 user@pserver
```

ここで -N はログインシェルとして起動しないようにするためのオプションです²。これがなくても通信そのものはできますが、コマンドを起動した端末を占有してしまいます。

次に -L オプションに続けて、ローカルのポート番号 (*localhost*)、サーバの名前 (*pserver*) とポート番号 (2401) を “:” (コロン) で区切って指定します。“2401” は CVS をサーバとして動かす時のポート番号です (2.3 節参照)。違うポートに変更した場合は、ここも同様に変更してください。ローカルポートは適当な番号を自分で指定して使います。

CVS クライアントが利用するポートを変更するには、環境変数 \$CVS_CLIENT_PORT を用いて以下のように指定します。

¹この暗号化は頑健な手法によるものではありません。暗号化されているから安全とは思わないでください。

²これは SSH2 のみに対応したオプションです。

```
% setenv CVS_CLIENT_PORT 10000
```

2.3 サーバ側の設定

ext 接続の場合、認証は CVS では行なわないので、サーバ側の準備は特に必要ありません。pserver 接続の場合、サーバ側にはいくつかの設定が必要です。まず、CVS サーバを `/etc/inetd.conf` に追加します。

```
2401 stream tcp nowait root /usr/local/bin/cvs cvs -f --allow-root=/usr/cvsroot pserver
```

ディレクトリは適宜変更してください。

コンパイル時に変更していない場合は“2401”がポート番号です。プログラムの場所は適宜変更してください。--allow-root には、レポジトリとしてアクセスを許可するディレクトリを書いてください。複数のディレクトリを指定する場合は、このオプションを繰り返し書きます。この後 `inetd` を再起動させるか、`kill -HUP` します。

上述のポート番号を直接 `inetd.conf` に書けない場合は `/etc/services` に

```
cvspserver 2401/tcp
```

と記述した上で、“2401”のかわりに“cvspserver”と書きます。

次に、この CVS サーバを使えるユーザを登録します。登録はユーザをファイルに追加することで行ないます。ファイルは `passwd` ファイルと似た形式で、例えば、以下のようになっています。

```
anonymous:
bach:ULtgRLXo7NRxs
melissa:tGX1fS8sun6rY:pubcvs
```

1行が1人分で、左から (CVS の) ユーザ名、(暗号化された) パスワード、利用者権限となっています。

ユーザ“anonymous”はパスワードがありませんので、誰でも CVT の利用を許可します。ユーザ名だけで、パスワードを指定しない場合は、匿名で読み込み専用でレポジトリにアクセスさせるためのものです。ソフトウェアの公開などに利用されます。

次の“bach”はパスワードを正しく入力した時のみ、CVS の利用が許可されます。“bach”は CVS のユーザ名です。利用者権限が明記されていないので、レポジトリを置いている計算機上にも同じ名前のユーザが必要で、レポジトリにはこのユーザの権限でアクセスします。

最後の“melissa”は、レポジトリを置いている計算機上の“pubcvs”というユーザで CVS のレポジトリにアクセスします。他の CVS ユーザに対しても、レポジトリを置いている計算機上の同じユーザを割りあてることができます。よって、レポジトリにアクセスできる権限の管理が簡単になります。つまり、レポジトリのある計算機上のユーザを“pubcvs”だけに制限すれば、“pubcvs”だけがレポジトリに対し書き込み権限を持てばよいことになります。

パスワードの暗号化は、標準的な Unix に付属の `crypt()` 関数によって暗号化されたものと同じです。したがって、なんらかの方法でパスワードを暗号化した文字列を作り、これをファ

イルに書く必要があります。例えば、Web サーバ Apache に付属する `htpasswd` などを利用するとよいでしょう。

`ext` 接続と `pserver` 接続のうち、レポジトリを提供する側からすると、`ext` 接続が準備が少なく簡単でしょう。しかし、`pserver` 接続の場合はパスワードをかけずに読み取りのみ許可した状態でレポジトリを公開することもできます。

2.4 ファイルの使用許可

レポジトリは過去の変更も含め、プロジェクトの全ての情報が格納されています。これを、誰でも変更できたり消去できるようにしておく、情報を改竄されたり、悪意を持って情報を消去されたりするかもしれません。しかし、複数の人間がプロジェクトに参加する場合、なんらかの方法で参加している人だけはファイルを書き換えたりする必要があります。

`ext` 接続で外部プログラムにより認証をする場合は、プロジェクトに参加している人はレポジトリのある計算機にログインが可能になっているはず。この場合、少なくともファイルの修正を行なうユーザは、レポジトリ中のディレクトリに書き込み可能である必要があります。つまり、プロジェクトごとにグループを作り、CVS のプロジェクト全体が格納されているディレクトリの所有グループとします。

`pserver` 接続の場合は、CVS のユーザ名に割りあてたユーザで同様のアクセスを管理します。

3 WinCVS

Windows 上で CVS を使うには、コマンドラインで動く CVS クライアント [1] を利用する方法があります。実行ファイルをパスの通った場所に置くだけで、UNIX 上での CVS と同様に使えます。

GUI 付きの CVS クライアントとしては、WinCvs [3] が有名です (図 1 参照)。インストール

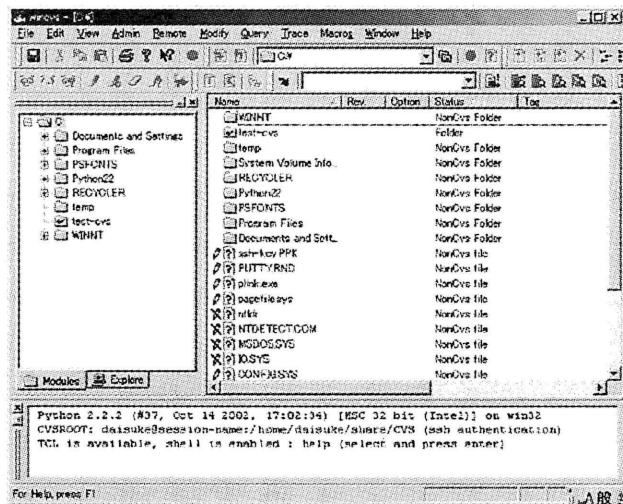


図 1: WinCvs の画面

は、ダウンロードしたファイルを展開するとできる `setup` ファイルをダブルクリックし、インストール先やライセンスの確認などに答えていくだけです。ここでは主にインストール後の設定や SSH と一緒に利用する方法を説明します。SSH との併用については、WinCvs のサイトにドキュメントが用意されていますので参照してください [4]。

WinCvs には安定版 (1.2) と開発版 (1.3) がありますが、ここでは開発版 (1.3b13) を用いて説明します³。動作確認は Windows 2000 と Windows XP で行ないました。

3.1 SSH を準備する

WinCvs でも ext 接続で認証に SSH を利用したり、pserver 接続の通信を SSH でポート転送して、安全に通信することが可能です。UNIX では、どちらの場合でも同じ `ssh` コマンドを利用できますが、Windows では少し事情が異なります。

ポート転送には TTSSH (TeraTerm に SSH の機能を付加したプログラム) や PuTTY が利用できます。また、ポート転送に特化したプログラム PortForwarder も利用できます。

一方、ext 接続する場合、Windows でも外部プログラムとして SSH を実装したコマンドを利用できます。しかし、SSH クライアントとしてよく用いられる TTSSH は、ポート転送機能はあるものの、ext 接続の外部プログラムとして利用できません。ext 接続の外部プログラムとしては、Windows で動作する OpenSSH や PuTTY を利用します。どちらも WinCvs から利用できますが、`ssh-agent` コマンドに相当する `pageant` が付属している PuTTY がよいでしょう。これを利用すると、`pageant` 起動時にパスワードを入力すれば、それ以降は WinCvs から CVS サーバへ何度接続しても、パスワードやパスワードを入力する必要はありません。

- ✓ (WinCvs に限らず) CVS では、各サブコマンドを実行するごとにパスワードの入力が必要になります。UNIX 環境で `ssh` と `ssh-agent` を併用する場合は、何度 SSH の通信が発生しても、パスワードの入力は最初の一度で済みます。これと同じことが PuTTY で可能になります。筆者が調べた限りでは、WinCvs と SSH を併用する場合、パスワードの管理までしてくれるのは PuTTY だけのようです。

PuTTY のインストールは、個々のコマンドごとにも可能です。ファイルは PuTTY のサイト [2] からダウンロードします。個々にインストールする場合は、鍵を生成する `puttygen`、`ssh-agent` に相当する `pageant`、他のソフトウェアから PuTTY の機能を利用するときの橋渡し役となる `plink`、それから PuTTY の本体である `putty` を適当なフォルダにコピーしてください。WinCvs で利用する前に、コマンドプロンプトからパスワードなしで CVS サーバへログインできることを確認しておいてください。ここでは PuTTY コマンドの詳しい使い方は省略します。

- ✓ PuTTY では、接続先のホスト名やユーザ名などをファイルに保存しておき、このファイルを引数として PuTTY を実行することができます。WinCvs で使う場合には、あらかじめ適当なファイル名でホスト名等を保存しておく必要があります。

3.2 WinCvs の起動と設定

WinCvs を起動する ([プログラム] → [WinCvs] → [WinCvs]) とちょっとした使い方のポイントが表示されます。他のポイントを見たい場合は [Next Tip] を、そうでなければ [Close] ボタ

³安定版では、後で説明する `pageant` を併用できませんでした。

ンを押します。次回から起動時にこれを表示しないようにするには [Show Tips on StartUp] のチェックをはずしてください。デフォルトでチェックされているので、毎回表示されます。

“Tip of the Day” 画面を閉じると、WinCvs を初めて起動した場合は図2のように、レポート

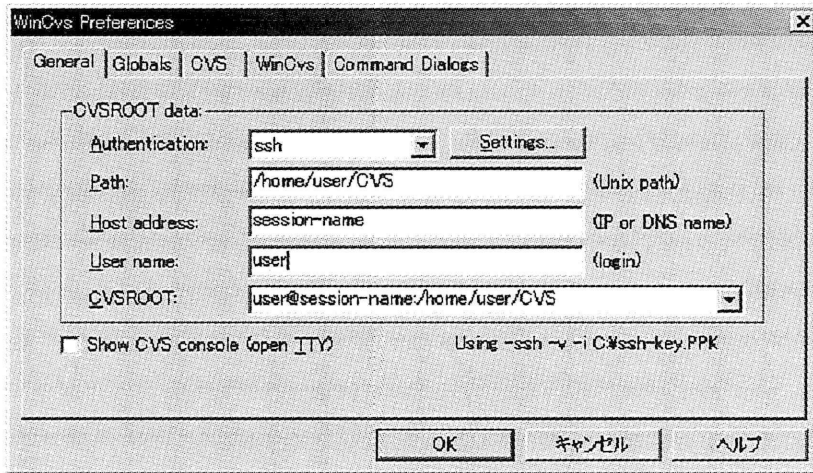


図2: レポジトリの入力画面

リの入力を求められます。

“Authentication:” は認証方法の指定で、外部コマンドで SSH を利用する場合は “ssh” を選択します。“Path:” と “User name:” は CVS サーバ上でのレポジトリの場所 (ディレクトリ) とユーザ名です。“Host address:” は、普通は CVS サーバを IP アドレスか名前前で指定します。しかし、pageant を利用する場合は PuTTY で CVS サーバに接続するときの情報を保存したファイル名を指定します。“CVSROOT:” は、上記の情報から自動的に生成されます。これを後から変更したい場合は [Admin] → [Preferences] の [General] タブで行ないます。

ssh による認証をする場合は “Authentication:” の右の “Settings...” ボタンを押し、図3の画面で SSH に関する設定を行ないます。

一番上の “RSA private key file (identity):” は、PuTTY 付属の puttygen で生成した秘密鍵ファイルを指定します。pageant を利用する場合には指定します。これを指定しないとパスワード認証になります。

次は実際に認証に使うコマンドの指定です。putty そのものではなく plink コマンドを指定します。

最後のオプションは “-ssh” と指定します。

これで WinCVS が使えるようになりました。認証の時に必要ですので、あらかじめ pageant を起動して、先程指定した秘密鍵を pageant に追加しておいてください。

レポジトリ以外は特に指定する必要はありません。指定する必要がでてきたときに、WinCvs 側から入力を求められます。また、これらの入力値は [Admin] → [Preferences] から変更可能です。

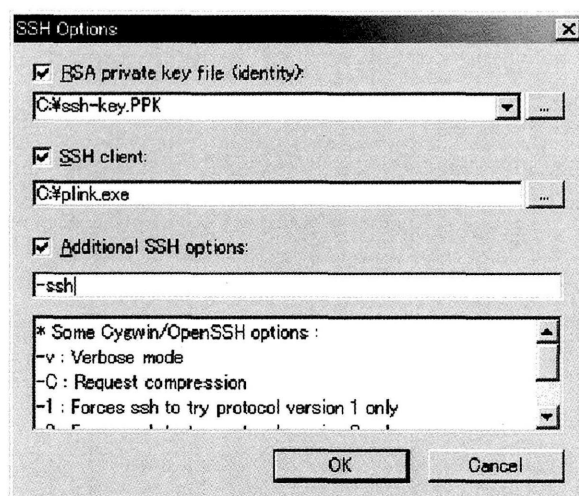


図 3: SSH の設定画面

3.3 WinCvs を利用する

作業用コピーを取り出すには [Create] → [Checkout module] とします。図 4 の上の空欄にプ

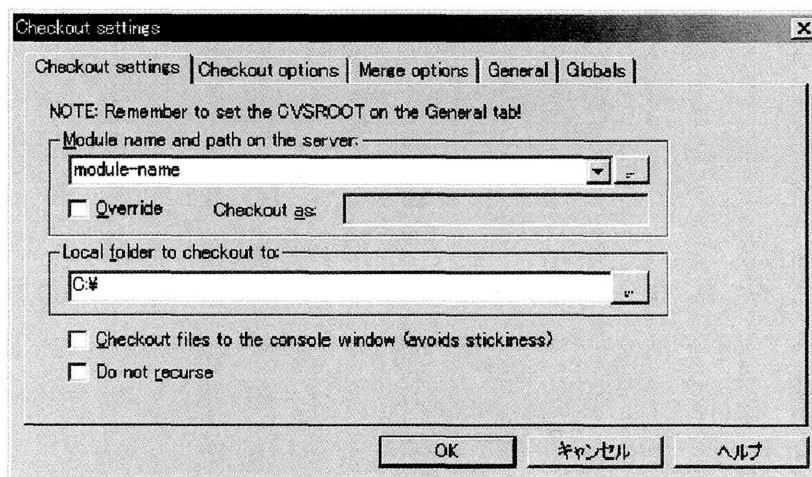


図 4: 作業用コピーの取り出し

ロジェクト名を、下の空欄にはどこに作業用コピーを置くかを指定します。

修正をレポジトリに反映させるには、対象となるファイルを選択して [Modify] → [Commit selection] とします。逆に他人が修正した最新のレポジトリの内容を作業用コピーに反映させるには、同じく [Modify] メニューから [Update selection] をクリックしてください。

4 Mac OS

この節では Mac OS で使える CVS クライアントを紹介します。Mac OS X は最初から CVS はインストールされていて、UNIX と同様にコマンドラインから使います。

GUI ベースのツールでは、WinCvs の配布元 [3] において MacCvs が提供されています。Mac OS X に対応した MacCvsX と Mac OS 8/9 対応の MacCvs があります。ここでは MacCvsX について説明します。動作確認は Mac OS X 10.2.6 で行ないました。

MacCvsX はアルファ版である 3.3 と、ベータ版である 3.2 と、安定版である 3.1 の系列があります。レポジトリをネットワーク上に置いて、サーバとの認証に SSH2 を利用したい場合は、ベータ版である MacCvsX-3.2 以降を利用します。ここでは MacCvs-3.3-alpha2 を使います。

配布形式は、ソースファイルとバイナリですが、ここではバイナリによるインストールの説明をします。MacCVS32b14.Mach0.sit というファイルをダウンロードし、Stuffit Expander などで展開します。展開したフォルダにある `maccvsX` が実行ファイルです。

インターフェイスは Mac OS X らしくなっていて (図 5 参照) ⁴、WinCvs のそれとは多少異なります。しかし、設定や使い方は WinCvs とほぼ同じです。

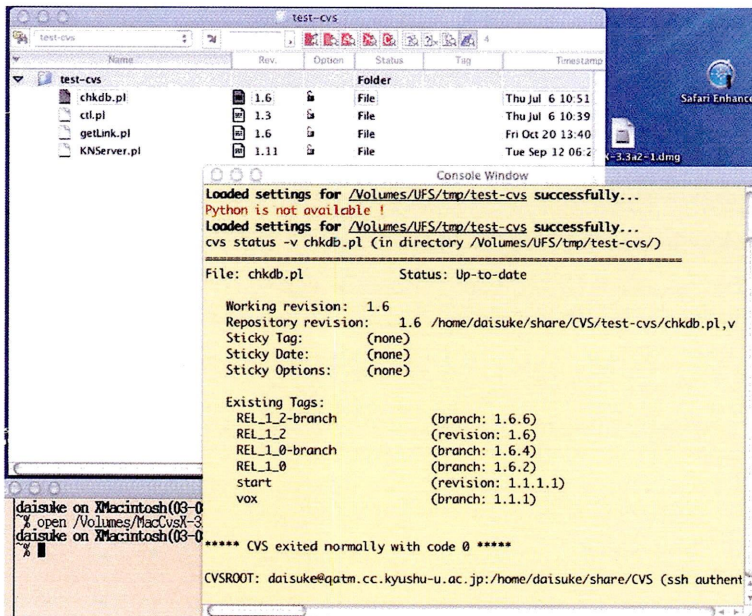


図 5: MacCVS の画面

SSH との併用については、Mac OS X にはコマンドラインから使える `ssh` コマンドが最初からインストールされていますので特に問題はありますが、MacCvsX を `ssh-agent` の配下にするにはログイン時に `ssh-agent` を起動しておき、自分のプロセスをすべて `ssh-agent` の配下にするか、`ssh-agent` を実行したシェルから `open` コマンドで以下のように MacCvsX を起動します。

⁴Classic なユーザにとっては Mac らしくはないかもしれませんが。

```
open /Volumes/MacCvsX-3.3a2-1/MacCvsX-3.3a2/MacCvsX.app
```

5 便利な機能

この節では CVS に関連した便利な機能を紹介します。

5.1 キーワード置換

作業ファイルに関する情報は `status` や `log` コマンドなどで見れますが、これらの情報をファイル中に埋め込む機能が、キーワード置換です。例えば、ファイルのリビジョン番号が埋め込まれていれば、以前のリビジョンとの差分を取るときなど、どのリビジョンと比較すればいいのかすぐに分かります。

埋め込む情報ごとにキーワードが決まっています。例えば、埋め込みたい情報のキーワードが “keyword” の場合にキーワード置換を利用するには、“\$keyword\$” と文字列をファイル中に書いておき、レポジトリに登録します。これで、次回レポジトリから作業用コピーを取り出すときに “\$keyword: value\$” という形式に変更されます。一度 “\$keyword: value\$” という形式に置換された文字列も、置換の対象ですので “value” の部分は適切に更新されます。

表 1 に、よく使うキーワードの一覧を載せておきます。

表 1: キーワード一覧

キーワード	説明
Author	このリビジョンを格納したユーザ名
ID	レポジトリのファイル名、リビジョン、日付、最終変更者、ロックしている人
Header	ID と同じだがファイル名がフルパスで表記される
Name	(タグ名を指定してファイルを取り出したときの) タグ名
Date	このリビジョンを格納した日付

5.2 貼り付いたタグ

貼り付いたタグ (sticky tag) は、通常の利用ではあまり気にする必要はありません。タグや日付けなどを指定して作業用コピーを取り出したときに、この作業用コピーにはタグや日付けが “貼り付いた” と表現されます。

貼り付いたタグのおかげで、この作業用コピーで作業する場合は、作業の対象はデフォルトで指定したタグや日付けに制限されます。つまり、これらのタグや日付けは checkout 時に指定すれば、その後の `commit` や `update` など他のサブコマンド利用時には指定する必要はありません。

貼り付いたタグの情報は `status` コマンドで表示されます。“SUBMITTED” というタグを指定して取り出したファイルの例です。

```

=====
File: cvs.tex           Status: Up-to-date

Working revision:      1.6
Repository revision:  1.6   /home/daisuke/share/CVS/Koho/cvs/cvs.tex,v
Sticky Tag:           SUBMITTED (revision: 1.6)
Sticky Date:          (none)
Sticky Options:       (none)

```

特定のファイルにだけタグを貼り付けることも可能です。この機能を利用すれば、他人が仮に変更を加えていたとしても、`update` で作業用コピーにその変更を取り込まないことができます。例えば、あるファイルに大規模な変更を加えて不安定になる場合などに有効でしょう。貼り付いたタグを剥がすには `cvs update -A` とします。

5.3 タグの変更

タグは履歴を記録するためのものであり、一般に変更する必要はなく、また、変更すべきではありません。しかし、タグの名前を間違えたときなど、タグを変更したいこともあります。

タグを削除するには `cvs tag -d TAGNAME` とします。`rtag` でも同様にできますが、この場合はモジュール名も指定します。その後、新しくタグを打てば変更したことになります。

また、タグの名前は同じでも、これが指すリビジョンを変更する方法もあります。例えば、`filename` で指定されたファイルのあるリビジョンに“TAG”というタグがついていて、このリビジョンを1.6に変更したいとします。この場合は、指したいリビジョンと-F オプションを指定して `cvs tag -r 1.6 -F TAG filename` とします。

5.4 レポジトリの移動

何らかの理由でレポジトリを移動しなければならない場合もあるでしょう。例えば、出張先などでネットワークにはつながらない場合を考えてみましょう。ノートパソコンは持っていますが、レポジトリは研究室のサーバにあります。出張中も CVS で管理するファイルのコピーを編集することは可能です。しかし、変更をレポジトリに格納できず、しかも、適当な区切りごとのタグも打てません。これでは CVS で管理している意味が半減してしまいます。このような場合、一時的にノートパソコンのディスク上にレポジトリを移せば、出張中も CVS が使えます。帰ってきたら、レポジトリを元のサーバに戻します。

レポジトリといっても単なるファイルですので、移動あるいはコピーすること自体は問題ありません。新しいレポジトリが決まれば、その後で取り出した作業用ファイルで作業することも問題ありません。したがって、レポジトリを移す前に、現在取り出している作業用コピーに施した変更点をすべてレポジトリに格納し、その後レポジトリを移します。そして作業用コピーを取り出せば、移動の影響はありません。

問題は、移動する前に取り出していた作業コピーが古いレポジトリを指している場合です。この作業用コピーが変更されていたとして、この変更を活かしたい場合は、直接このディレクトリにある `CVS/Repository` と `CVS/Root` を書きかえます。Root には `$CVSROOT` の情報が、`Repository` にはレポジトリのファイル名が絶対パスで書いてあります。

参考文献

- [1] Concurrent Versions System - The open standard for version control <http://www.cvshome.org>
- [2] PuTTY: A Free Win32 Telnet/SSH Client <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
- [3] CvsGui <http://www.wincvs.org/>
- [4] SSH with WinCVS <http://www.wincvs.org/ssh.html>
- [5] 池田 大輔、CVS でファイル管理、九州大学情報基盤センター広報 全国共同利用版 Vol. 3, No. 1, pp. 14-28, 2003.

平成 15 年度講習会開催一覧

講習会名	開催日(曜日)	受講者数	備考
UNIX 初級	5 月 13 日 (火)	47 名	
UNIX 初級	5 月 14 日 (水)	23 名	
プログラミング言語利用	5 月 16 日 (金)	36 名	
プログラミング言語利用	5 月 19 日 (月)	21 名	
UNIX 中級	5 月 23 日 (金)	26 名	
UNIX 中級	5 月 27 日 (火)	33 名	
スーパーコンピュータ VPP 並列プログラミング	5 月 29 日 (木) ～ 30 日 (金)	15 名	
α -FLOW	6 月 6 日 (金)	4 名	
MOPAC2000	6 月 12 日 (木)	5 名	
AVS	6 月 19 日 (木) ～ 20 日 (金)	12 名	
VisLink	6 月 20 日 (金)	6 名	
MASPHYC	6 月 27 日 (金)	8 名	
汎用 UNIX サーバ GP7000F 並列プログラミング	7 月 3 日 (木) ～ 4 日 (金)	9 名	
LS-DYNA	7 月 11 日 (金)	7 名	
Gaussian98	7 月 14 日 (月)	18 名	
MSC.Nastran / MSC.Patran	7 月 24 日 (木)	13 名	
MSC.Marc / MSC.Mentat	7 月 25 日 (金)	9 名	
スカラー並列サーバ プログラミング	7 月 30 日 (水)	7 名	
Tecplot	8 月 7 日 (木)	2 名	
SpaceFinder	8 月 18 日 (月)	9 名	

UNIX 初級
受講者内訳

5月13日(火)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	2	20	23	2	47	
合計	0	0	0	0	2	20	23	2	47	

UNIX 初級
受講者内訳

5月14日(水)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	1	4	10	6	2	23	
合計	0	0	0	1	4	10	6	2	23	

プログラミング言語利用
受講者内訳

5月16日(金)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	2	17	16	1	36	
合計	0	0	0	0	2	17	16	1	36	

プログラミング言語利用

受講者内訳

5月19日(月)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	4	11	5	1	21	
合計	0	0	0	0	4	11	5	1	21	

UNIX 中級

受講者内訳

5月23日(金)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	2	14	9	1	26	
合計	0	0	0	0	2	14	9	1	26	

UNIX 中級

受講者内訳

5月27日(火)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	2	5	15	8	3	33	
合計	0	0	0	2	5	15	8	3	33	

スーパーコンピュータ VPP 並列プログラミング

受講者内訳

5月29日(木)～30日(金)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	1	0	0	2	3	2	4	0	12	
広島大学	0	1	0	0	0	0	0	0	1	
九州工業大学	0	0	0	1	1	0	0	0	2	
合計	1	1	0	3	4	2	4	0	15	

α-FLOW

受講者内訳

6月6日(金)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	2	0	2	0	4	
合計	0	0	0	0	2	0	2	0	4	

MOPAC2000

受講者内訳

6月12日(木)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	1	1	2	1	0	5	
合計	0	0	0	1	1	2	1	0	5	

汎用可視化システム AVS

受講者内訳

6月19日(木)～20日(金)実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	3	5	4	0	12	
合計	0	0	0	0	3	5	4	0	12	

スパコンコンピュータ連携ツール VisLink

受講者内訳

6月20日(金) 実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	1	1	4	0	6	
合計	0	0	0	0	1	1	4	0	6	

MASPHYC

受講者内訳

6月27日(金) 実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	2	0	1	2	1	2	0	8	
合計	0	2	0	1	2	1	2	0	8	

汎用 UNIX サーバ GP7000F 並列プログラミング

受講者内訳

7月3日(木)～4日(金) 実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	1	0	3	4	1	0	9	
合計	0	0	1	0	3	4	1	0	9	

LS-DYNA

受講者内訳

7月11日(金) 実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	0	3	4	0	0	7	
合計	0	0	0	0	3	4	0	0	7	

Gaussian98

受講者内訳

7月14日(月) 実施

職名 学校名	職名							学部・ 研究生	その他 (技官・ 事務官)	合計
	教授	助教授	講師	助手	博士	修士				
九州大学	0	0	0	2	6	4	6	0	18	
合計	0	0	0	2	6	4	6	0	18	

MSC.Nastran / MSC.Patran

受講者内訳

7月24日(木)実施

職名 学校名	職名						学部・ 研究生	その他 (技官・ 事務官)	合 計
	教授	助教授	講師	助手	博士	修士			
九州大学	0	0	0	0	3	7	3	0	13
合 計	0	0	0	0	3	7	3	0	13

MSC.Marc / MSC.Mentat

受講者内訳

7月25日(金)実施

職名 学校名	職名						学部・ 研究生	その他 (技官・ 事務官)	合 計
	教授	助教授	講師	助手	博士	修士			
九州大学	0	0	0	0	1	6	1	0	8
有明高専	0	0	1	0	0	0	0	0	1
合 計	0	0	1	0	1	6	1	0	9

スカラー並列サーバプログラミング

受講者内訳

7月30日(水)実施

職名 学校名	職名						学部・ 研究生	その他 (技官・ 事務官)	合 計
	教授	助教授	講師	助手	博士	修士			
九州大学	0	0	0	0	3	3	1	0	7
合 計	0	0	0	0	3	3	1	0	7

Tecplot

受講者内訳

8月 7日 (木) 実施

職 名 学校名	職 名						学部・ 研究生	その他 (技官・ 事務官)	合 計
	教授	助教授	講師	助手	博士	修士			
九州大学	0	0	0	0	0	2	0	0	2
合 計	0	0	0	0	0	2	0	0	2

SpaceFinder

受講者内訳

8月18日 (月) 実施

職 名 学校名	職 名						学部・ 研究生	その他 (技官・ 事務官)	合 計
	教授	助教授	講師	助手	博士	修士			
九州大学	0	0	0	0	2	5	2	0	9
合 計	0	0	0	0	2	5	2	0	9

お 知 ら せ

Fortran コンパイラ一部障害のお知らせ (VPP5000/64)	38
Fortran コンパイラ一部障害のお知らせ (GP7000F/900)	39
九州大学の対外接続ネットワークの一時停止について	41
年末・年始計算サービスの一時停止のお知らせ	41
科学研究費及び産学連携等研究費での利用期限について	42

◆ Fortran コンパイラ一部障害のお知らせ (VPP5000/64)

スーパーコンピュータVPP5000/64(ホスト名: kyu-vpp)の Fortran コンパイラに一部障害があったことが判明しました。内容、発生条件は下記の通りです。障害は各条件をすべて満たした場合に発生する可能性のあるもので、条件のうち一つでも該当箇所がない場合には発生しません。障害は8月13日(水)に修正済です。

なお汎用UNIXサーバー(kyu-cc)、スカラ並列サーバー(kyu-ss)では障害は発生しません。また、修正完了後、障害回避のため、発生条件に該当する翻訳時オプションを指定して作成したオブジェクトファイル、実行可能ファイルの再作成をお願いします。

《発生条件》以下の条件を満たす場合、計算結果に誤りが生じる場合がある。

1. 多重ループが存在する
2. 最内ループに構造型が存在する
3. 2.の構造型要素に配列が存在する
4. 1.のループの一つのDO文の初期値または終値に配列要素が存在する
5. 4.のループの外側にループが存在する
6. 4.と5.のDO文の間に単純変数への定義が存在する
7. 6.の単純変数が4.の配列の添字に使用されている
8. 6.はベクトル化されている

《プログラム例》

```

integer * 4 mm(10)
integer :: n1, m1, m2, CO, SP
parameter (n1=1, m1=11, m2=11, CO=3, SP=10)
type vec
  real(8) :: b(n1:m1, n1:m2)
end type
type(vec) :: re(CO, SP)

data mm/1, 2, 3, 4, 5, 1, 2, 3, 4, 5/
zzz=3.
do i=n1, m2

```

```

do j=n1,m2
do k=1,SP
do l=1,CO
  re(l,k)%b(i,j) = 1.
enddo
enddo
enddo
enddo
!ocl nounroll
do i=1,10
  id = i
  do j=mm(id),mm(id)+5
    do i1=1,SP
    do i2=1,CO
      re(i2,i1)%b(j,i) = re(i2,i1)%b(j,i)+zzz ! 条件 2)および3)に該当
    enddo
    enddo
    enddo
    enddo
  write(6,33) re(1,1)%b(1,1)
33  format('ans = ',5f10.5)
end

```

《正しい結果》

ans = 4.00000

《誤った結果》

ans = 1.00000

(ソフトウェア室 092-642-2295)
e-mail: soft-room@cc.kyushu-u.ac.jp

◆ Fortran コンパイラ一部障害のお知らせ (GP7000F/900)

汎用UNIXサーバGP7000F/900(ホスト名: kyu-cc)のFortran コンパイラに一部障害があったことが判明しました。内容、発生条件は下記の通りです。障害は各条件をすべて満たした場合に発生する可能性のあるもので、条件のうち一つでも該当箇所がない場合には発生しません。障害は9月24日(水)に修正済です。

障害に該当する可能性は、-O4 以上の最適化レベルを指定した場合に発生するもので、省略値オプションまたは-Kfastオプションは該当しません。また、修正完了後、障害回避のため、発生条件に該当する翻訳時オプションを指定して作成し

たオブジェクトファイル, 実行可能ファイルの再作成をお願いします.

《発生条件》以下の条件を満たす場合, 翻訳時異常終了する, または, 実行結果異常となる場合がある.

1. -O4以上の最適化オプションが指定されている.
2. -kparallelオプションが指定されていない.
3. 3重以内のタイトな(同じネストレベルのループが複数存在しない)多重ループが存在する.
4. 3. のループにおいて最内にのみ実行文が存在する.
5. 4. の実行文において複素数型配列への参照または代入が存在する.
6. 最内ループの増分値が1である.

《プログラム例と実行例》

```

parameter (n=10000)
complex*16 x(0:n-1, 3, 4), w(0:n-1, 3, 4)
logical mask_e(0:n-1)

w=(0, 0)
x=(1, 2)
do k=0, 10000
  if (k.LT.5000) then
    mask_e(k)=.TRUE.
  else
    mask_e(k)=.FALSE.
  endif
enddo

do j=1, 4
do i=1, 3
do k=0, n-1
  if (mask_e(k)) then
    w(k, i, j) = x(k, i, j)
  end if
end do
end do
end do
print*, (mask_e(k), k=4999, 5000)
print*, (w(k, 4, 3), k=4999, 5000)
do j=1, 4
do i=1, 3
do k=0, n-1

```

! この3重ループが条件3と4に該当

! 条件6に該当

! 配列wとxが条件5に該当

```

    if (mask_e(k)) then
      if (w(k,i,j).NE.x(k,i,j)) then
        stop 'NG'
      end if
    else
      if (w(k,i,j).NE.(0,0)) then
        stop 'NG'
      end if
    end if
  end do
end do
end do
print*, "OK"
end

```

<誤った結果>

```

t f
(0.0000000000000000e+00,0.0000000000000000e+00)
(0.0000000000000000e+00,0.0000000000000000e+00)
jwe0002i stop NG

```

<正しい結果>

```

t f
(1.0000000000000000,2.0000000000000000)
(0.0000000000000000e+00,0.0000000000000000e+00)
OK

```

(ソフトウェア室 092-642-2295)
e-mail: soft-room@cc.kyushu-u.ac.jp

◆九州大学の対外接続ネットワークの一時停止について

スーパーSINET九州大学ノードの一時停止に伴い、下記の日時に九州大学はネットワーク対外通信が不通になります。不通の時間帯も計算サービスを行ないませんが、九州大学外のネットワークからの利用はできませんのでご了承ください。

記

日時：平成15年12月14日(日) 9時～14時

(システム管理掛 092-642-2308)

◆年末・年始計算サービスの一時停止のお知らせ

年末・年始計算サービスの一時停止のお知らせ

年末・年始の期間、下記のとおり計算サービスを一時停止します。

システム名	停止期間	サービス開始日
スーパーコンピュータ (VPP5000/64) ホスト名:kyu-vpp	平成15年12月30日(火)17時 から 平成16年1月5日(月)17時	平成16年1月5日(月)17時
汎用UNIXサーバ (GP7000Fモデル900) ホスト名:kyu-cc		
スカラー並列サーバ (GS320) ホスト名:kyu-ss		
端末サーバ ホスト名:wisdom	停止しません	

*センター内オープン機器室及び情報サロンは12月26日(金)12時から1月4日(日)まで閉館します。

(システム運用掛 ダイヤルイン092-642-2307)

◆科学研究費及び産学連携等研究費での利用期限について

科学研究費及び産学連携等研究費での最終利用期限は、平成16年2月11日(水)です。

なお、1月～2月11日分は2月中旬に請求処理を行いますので、2月上旬までに年間使用額を確定されたい方は、利用期限を平成15年12月31日としてください。
既に登録が済んでいる場合は、所属センター登録の方は、12月26日までに「計算機利用申請書」で、第2センター登録の方は、所属センターの第2センター登録用コマンドで利用期限の変更を行ってください。

確定額について、電話等による利用金額のお問い合わせは受けられませんので、必ず計算機利用通知書でご確認くださいませうお願いします。

(共同利用掛 092-642-2305)

人 事 異 動

◎転出・退職

発令年月日	異動後の官職等	氏 名	異動区分	異動前の官職等
15.11.30	国立情報学研究所 教授	三浦 謙一	辞 任	客員教授

◎転入等

発令年月日	異動後の官職等	氏 名	異動区分	異動前の官職等
15. 8. 1	ネットワーク運用掛主任	多久島 智	転 任	大学評価・学位授与機構 管理部情報課ネットワーク管理 係主任
15.10. 1	システム管理掛	亀 岡 謙 一	採 用	
15.11.9	システム運用掛	高 木 早智子	育児休業	
15.11.10	システム運用掛	黒 木 ささら	臨時的任用	

編 集 後 記

編集後記の執筆は、広報関係者の間で順番にまわしています。一度に2人しか書かないことと、関係者の人数がある程度多いこともあって、忘れたころに順番がやってきます。が、前回書いた時もだいたい同じ時期に書いたような気がします。
「スポーツの秋を満喫している」と書いた記憶があるからです。編集後記を担当の人にメールで送ったはずなので、過去のメールを検索してみると、ありました。だいたい同じ時期に書いてます。今年のこの時期は、例年より忙しくなっていますが、やっぱりスポーツの時間を頑張って確保しています。

Yr ydwyf fi'n dysgu Cymraeg.

「私はウェールズ語を勉強しています」というウェールズ語の文です。ウェールズ語はいわゆる少数言語の1つで、法政策などの面ではかなり恵まれた少数言語です。しかし、いざ勉強を始めようとテキストや辞書を探してみると、残念ながらあまり選択の余地がありません。一方、専門用語や略称のほとんどが英語という、英語どっぶりのコンピュータ・ネットワークでは、もちろん英語資源が爆発的に日々増加していますが、その陰でしっかりこれら少数言語資源も少しずつですが充実してきているようです。今では日本にいながらにしてウェールズ語のニュースを読む、音声を聴く、動画を見聞きすることまでできます。とても心強く感じるのと同時に、このような流れが今後も大きくなることを願うばかりです。最後に僕が勉強しているウェールズ語のテキストに出てきたお気に入りの一文を紹介します。

Peidiwch â phoeni. Gadewch i'r athro barhau â'r ddarlith.

「心配するな。教授には講義を続けさせておきなさい」

(S.T.)

投稿のしおり

九州大学情報基盤センターでは、利用者の方々との深い交流と有益な情報交換のため、以下の種類の原稿を募集しています。

- 随想
- 計算機を利用した研究・開発の紹介
- アプリケーションの実用例や解説
- プログラミングの実例や解説
- センターに対する質問・要望
- 利用者の声
- 計算機やネットワークに関すること

執筆の際は、後述の《執筆上の注意》を必ずご参照ください。投稿された原稿の掲載については、広報委員会にて検討させていただきます。

別刷りは、原稿提出時に希望があれば、50部まで無料で差し上げます。50部以上希望される場合は、原稿提出時にご相談ください。

なお、投稿に関するお問合せは共同利用掛(ダイヤル092-642-2305)までお願いします。

《原稿送付先》

原稿の送付先は、以下のとおりです。

〒812-8581
福岡市東区箱崎6丁目10番1号
九州大学 情報基盤センター
共同利用掛 行

《ㄥ切》

原稿のㄥ切は、5・9・1月の15日です。発行は、その2ヶ月後になります。ㄥ切を過ぎた原稿は、その次の号への投稿扱いとなることがあります。

《執筆上の注意》

1. 原稿は原則としてワードプロセッサ等の出力結果とします。

用紙サイズはB5のみとし、書式は図1をご覧ください。手書きでも結構ですが、その場合センター規定の原稿用紙を使用してください。用紙は、共同利用掛にあります。フロッピーディスクや電子メールでの投稿は別途、共同利用掛までご相談ください。

2. ワードプロセッサによる出力は、写真製版とさせていただきますので、投稿原稿の品質にはご注意ください。
3. 手書きの場合は、黒鉛筆、黒ボールペンまたは黒インクのいずれかで書いてください。また、数字、英文字、大(小)文字、上(下)付き文字等混合しやすい文字ははっきりわかりやすく指定してください。
4. 原則として、常用漢字、現代かな使いで統一してください。
5. 第一ページには、必ず題名、著者名、所属および電子メールアドレスを記入してください。記入位置については、図1をご覧ください。

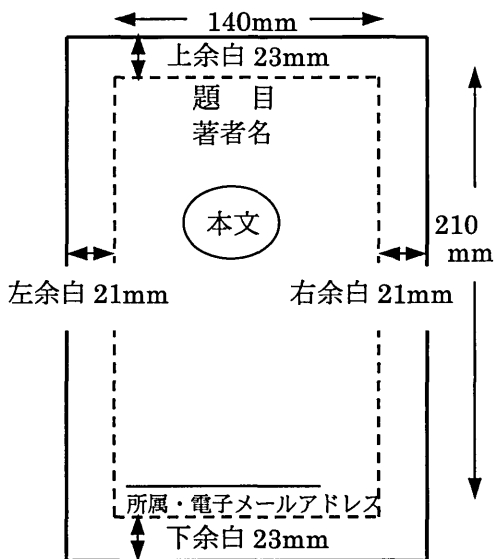


図1：書式設定

業 務 案 内

平成13年1月1日より

システム利用	サービス時間	各種保守作業による停止時間
スーパーコンピュータ 汎用UNIXサーバ	原則通年24時間 ただし、右欄保守作業の 停止時間を除く	◇定期保守 毎週水曜日 5時～12時30分
スカラー並列サーバ その他のコンピュータ	原則通年24時間	—

センター内利用	曜日	開始時刻	終了時刻
オープン利用機器	月～火	9:00	21:00
	水	12:30	
	木～金	9:00	

- 備考 1. 毎週水曜日は、計算機システムの保守のため12時30分から計算サービスを開始します。
 2. 16時45分以降（月～金）と土曜日及び日曜祝日は、無人運転を行っており、システム障害が発生した場合は、約20分後に自動的に計算サービスを再開します。
 ただし、30分経過後に再開できない場合は、計算サービスを打ち切らせていただきます。
 3. 臨時の停止、運用を行う場合は、その都度センターニュース等でお知らせします。

センター利用案内

092(642)内線番号

※は092(641)3131(トーン信号)内線番号

担 当 掛	階	業 務 案 内	内 線
研 究 部	6階	1. 計算機システムに関すること 2. ソフトウェアに関すること 3. データベースに関すること 4. 開発課題に関すること	2296
庶 務 掛	5階	1. 各種委員会に関すること 2. センター案内、見学に関すること	2303 8226※
会 計 掛		1. 会計一般に関すること	2304
共 同 利 用 掛 (第七地区協)	5階	1. 利用申請、登録に関すること 2. 利用の手引き、広報などの配付に関すること 3. 講習会に関すること 4. プログラム相談に関すること 5. 利用負担金に関すること 6. 出張利用者旅費に関すること 7. 他センター利用の手続に関すること 8. 各大型センター及び連絡所の調整に関すること 9. 連絡所の登録、変更及び廃止に関すること 10. センターへの要望、問い合わせに関すること	2305 8229※
図 書 室	4階	1. 図書・マニュアル及びセンター関係資料の保管、 閲覧及び貸出について	8237※
システム運用掛		1. 計算サービス全般に関すること 2. 計算機システムの運用に関すること	2307 8231※
システム管理掛		1. オペレーティングシステムに関すること 2. 計算機システムの管理に関すること	2308 8232※
ネットワーク運用掛		1. ネットワークの運用に関すること	4032
ネットワーク管理掛		1. ネットワークの管理に関すること	2309
業 務 受 付		2階	1. オープン室の利用に関すること 2. センターの利用に関すること
プログラム相談室		1. プログラム相談に関する相談・指導	2312

◇センター利用に関する質問・要望等の e-mail は下記のアドレスで受け付けます。ご利用ください。

e-mail アドレス : request@cc.kyushu-u.ac.jp

九州大学情報基盤センター広報
Vol. 3, No. 3
平成 15 年 11 月 発行
編集 九州大学情報基盤センター
広報委員会
印刷 松隈印刷株式会社